[redacted]

UIUCDCS-R-79-978 UILU-ENG 79 1726

Complete Problems in the First-Order Predicate Calculus

by

David A. Plaisted

July 1979

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

Complete Problems in the First-Order Predicate Calculus

by

David A. Plaisted
Department of Computer Science
University of Illinois
Digital Computer Lab
Urbana, IL 61801

Key words: Satisfiability, predicate calculus, resolution theorem proving, complete problems, computational complexity.

This research was partially supported by the National Science Foundation under grant MCS 77-22830.

Abstract

We present a survey of recent results concerning the complexity of deciding if a first-order predicate calculus formula in Schönfinkel-Bernays form is satisfiable. We also present results concerning the complexity of deciding the existence of resolution proofs of restricted depths and sizes. In this way we obtain natural problems complete for the classes NP, PSPACE, deterministic and nondeterministic exponential time, and deterministic and nondeterministic double exponential time. The results concerning the existence of resolution proofs of restricted depths and sizes seem to have implications for the design and analysis of resolution theorem proving programs. This report mainly lists results without proof.

We exhibit problems involving first-order predicate calculus formulae that are complete for the classes NLOGSPACE, P, NP, PSPACE, DEXPTIME, and NEXPTIME. These problems involve satisfiability and resolution proof depth for formulae in Schönfinkel-Bernays form (that is, the prefix is of the form $\exists\exists..\exists\forall\forall..\ \forall$). Most of these results are obtained by direct encoding of Turing computations, using an economical representation of the successor relation on integers. Some of the results are obtained by different methods. This work was originally motivated by a problem involving relational databases posed by Yehoshua Sagiv. It was then realized that this problem was related to recent work of Harry Lewis [7] involving decidable subclasses of the first-order predicate calculus. We improved his lower bound of nondeterministic time $c^{\sqrt{n}}$ for the Schönfinkel-Bernays class to nondeterministic time $c^{n/\log n}$. Harry Lewis has improved this to c^n since then [8].

We have also obtained some results which precisely characterize the difficulty of resolution theorem proving in the first-order predicate calculus (not restricted to Schönfinkel-Bernays form). In particular, one of these results states that it is complete for nondeterministic exponential time to decide if a depth d proof of a particular clause exists from a set W of clauses, if d is given in unary notation. Other results deal with the special case in which W consists entirely of 2-literal clauses. Also, it is NP-complete to determine if a size d resolution proof of a particular clause from a set W of propositional clauses exists, if d is given in unary. (This last result is not just a restatement of the NP-completeness of satisfiability of Boolean expressions.)

Some of the results for deterministic and nondeterministic exponential time completeness are obtained by encoding Turing computations using statements of the form "The symbol on tape square i at time j is s" or "The Turing machine at time j is at tape square i in state t." In this way we encode deterministic and nondeterministic Turing computations. Some of the results for polynomial space completeness are obtained by letting $P(x_0, x_1, \ldots, x_n)$ represent a configuration $x_0 x_1 \ldots x_n$ of a Turing machine, specifying the tape symbols, the current state, and the position of the read-write head. By cycling the variables to the left or the right, we get the effect of moving the read-write head to the right or to the left.

We first consider formulae of the classical first-order predicate calculus without the identity sign or function signs. In particular, an <u>atom</u> is a predicate symbol followed by a list of variables (and constants). A formula is said to be in <u>Schönfinkel-Bernays form</u> if it is of the form $\exists y_1 \exists y_2 \ldots \exists y_n \forall x_1 \forall x_2 \ldots \forall x_m A$ where A is a well-formed expression containing atoms and the Boolean connectives $\wedge$(conjunction), $\vee$(disjunction), $\equiv$(equivalence), and $\neg$(negation). We will assume that formulae in Schönfinkel-Bernays form have no free variables. The Skölemized (or functional) form of A is obtained by replacing each y_i in A by a distinct constant symbol, and by deleting all quantifiers. For example, $\exists y_1 \exists y_2 \forall x_1 \forall x_2 (P(x_1, y_1) \vee P(x_2, y_2))$ is a formula in Schönfinkel-Bernays form and its Skölemized form is $P(x_1, c_1) \vee P(x_2, c_2)$. It is known that a formula F is satisfiable iff its Skölemized form is satisfiable. We will assume all formulae

are in Skölemized form. Also, we will assume all formulae are expressed as a conjunction of disjunctions of literals, where a literal L is an expression of the form B or ¬B for some atom B. In the former case, we say L is a positive literal, and in the latter case we say L is a negative literal. A clause is a disjunction of literals. Hence we will assume all formulae are conjunctions of clauses. We say a problem S is complete for a class of problems if S is in the class and if every other problem in the class can be reduced to it by deterministic log tape reductions. Completeness for DLOGSPACE is defined by using deterministic one-way log-tape reductions as defined in [2]. Note that a problem may be complete for DEXPTIME under this definition even if it can be solved in time $c^{n/\log n}$.

We now give a list of results. We need more terminology. A Horn clause is a clause that has at most one positive literal. The positive literal of a Horn clause (if it exists) is called the consequent of the clause. The negative literals (if they exist) are called antecedents of the clause. We say two literals L1 and L2 are unifiable if they have a common instance. For Schönfinkel-Bernays form formulae, this means that there is some literal L that can be obtained from both L1 and L2 by replacing variables by constant symbols. Thus P(x, c) and P(d, x) are unifiable because they have P(d, c) as a common instance. However, P(x, x) and P(c, d) are not unifiable, nor are P(c, x) and P(d, y).

We use the notation $\#P \leq c$ to indicate that the number of predicate symbols in a class of formulae is bounded. Also, HORN indicates that all formulae in the class are conjunctions of HORN clauses. The notation DET means that no two consequents of distinct Horn clauses in the formula are unifiable, and that every (universally quantified) variable appearing in an antecedent must also appear in the consequent.

In particular, a clause having no consequent must have no universally quantified variables. This condition implies the determinism of a certain kind of program expressed as a set of Horn clauses, hence the notation DET.

Furthermore, 3LIT means that all formulae are conjunctions of clauses having three or fewer literals, and 2LIT means all clauses have two or fewer literals. The symbol E refers to the number of existential quantifiers and U refers to the number of universal quantifiers. With each combination of conditions, we indicate which deterministic or nondeterministic time or tape complexity class the problem is complete for. Note that $E = 2$ is equivalent to $E \geq 2$ for completeness results.

1. Complexity of the satisfiability problem for formulae in Schönfinkel-Bernays form with $E = 2$ and $\#P \leq c$.

Restrictions	Complete for
a) 3LIT	NEXPTIME
b) 3LIT HORN	DEXPTIME
c) 3LIT HORN, DET	PSPACE
d) 2LIT	PSPACE
e) 2LIT HORN	PSPACE
f) 2LIT HORN, DET	PSPACE

Comments: 1a) is due to Lewis [7]. 1d) can be obtained from the function generation problem of Kozen [6], but we have a simple direct reduction from Turing computations.

2. Complexity of the satisfiability problem for formulae in Schönfinkel-Bernays form with $E = 2$ and $\#P \leq c$ and $U \leq \log_2|W|$ where $|W|$ is the length of the formula W.

a)	3LIT	NP
b)	3LIT HORN	P
c)	3LIT HORN, DET	NLOGSPACE
d)	2LIT	CoNLOGSPACE
e)	2LIT HORN	CoNLOGSPACE
f)	2LIT HORN, DET	DLOGSPACE

Comments: All of these results can be obtained for $E = 2$, $\#P \leq c$, and $U = 0$ by encoding n predicate symbols as sequences of logn bits and using the results in Part 3.

3. Complexity of the satisfiability problem for propositional calculus formulae. This corresponds to Schönfinkel-Bernays form with $E = 0$ or $E = 1$. These results are all known or follow easily from known results.

a)	3LIT	NP
b)	3LIT HORN	P
c)	3LIT HORN, DET	NLOGSPACE
d)	2LIT	CoNLOGSPACE
e)	2LIT HORN	CoNLOGSPACE
f)	2LIT HORN, DET	DLOGSPACE
g)	$\#P \leq c$	Regular set

Comments: 3a) is of course Cook's result [1]. 3b) is essentially the complement of the GEN problem [4]. 3c) can be obtained from the graph accessibility problem GAP of [3]. 3d) is essentially from [3]. 3e) can be obtained from the complement of GAP of [3]. 3f) can be obtained from the GAP1 problem of [2].

4. Complexity of deciding the existence of various types of resolution proofs having depths or sizes restricted in various ways. The size of a resolution proof is the number of resolutions in it. Assume we are given an integer n and a formula W in Schönfinkel-Bernays form with E = 2 and #P $\leq$ c, and are looking for a resolution proof of NIL (the empty clause) from W. An all-positive resolution proof is a resolution proof in which one of the parent clauses in each resolution consists entirely of positive literals. In a unit resolution proof, one of the parents in each resolution must consist of a single literal. Note that all-positive resolution and hyper-resolution are identical for 2-literal Horn clauses.

a)	3LIT HORN, n in binary	Complete for DEXPTIME to decide if an (all-positive, unit, hyper-resolution) proof of (depth, size) n exists.
b)	3LIT HORN n in unary	PSPACE - complete to decide if an (all-positive, unit, hyper-resolution) proof of depth n exists. NP-complete to determine if an (all-positive, unit, hyper-resolution) proof of size n exists. (See 4f.)
c)	3LIT HORN, DET n in binary	PSPACE - complete to decide if a hyper-resolution proof of depth n exists. PSPACE - hard to decide if an (all-positive, unit) proof of depth n exists.
d)	3LIT HORN, DET n unary	CoNP - complete to decide if a hyper-resolution proof of depth $\leq$ n exists. CoNP - hard to decide if an (all-positive, unit) proof of depth $\leq$ n exists.
e)	2LIT HORN n binary	PSPACE - complete to decide if an (all-positive, unit) proof of (depth, size) n exists.
f)	2LIT HORN n unary	NP-complete to decide if an (all-positive, unit) proof of (depth, size) n exists. PSPACE - complete to decide if an arbitrary resolution proof of depth n exists.
g)	2LIT HORN, DET n binary	PSPACE- complete to decide if an (all-positive, unit) proof of (depth, size) n exists.

h) 2LIT HORN, DET n unary — P - complete to decide if an (all-positive, unit) proof of (depth, size) n exists. PSPACE - complete to decide if an arbitrary resolution proof of depth n exists.

Comments: 4b) is obtained by reduction from satisfiability of quantified Boolean formulae [11]. 4f), first part, is obtained by reduction from conjunctive normal form satisfiability. We do not know the complexity of 4d) if we require the proof to have depth equal to n. Many of these results can be extended to the problem of determining if there exists an inconsistent set of n ground instances of the clauses in W.

5. Complexity of deciding the existence of various types of resolution proofs of NIL from a set of clauses in the propositional calculus. (This corresponds to Schönfinkel-Bernays form formulae with E = 0 or E = 1.) Assume we are given a set W of clauses and an integer n.

a) arbitrary set of clauses, n unary — P-complete to decide if a unit resolution proof of depth n exists. NP-complete to decide if an (all-positive, hyper-resolution, arbitrary) proof of size n exists. NP-hard to decide if an (all-positive, arbitrary) proof of depth n exists. These results still hold if 3LIT is required.

b) 3LIT HORN, n unary — NP-complete to decide if an (all-negative, arbitrary) proof of size n exists. NP-hard to decide if an (all-negative, arbitrary) proof of depth n exists. P-complete to decide if an (all-positive, unit, hyper-resolution) proof of (depth, size) n exists.

c) 3LIT HORN, DET, n unary — CoNLOGSPACE - complete to determine if a hyper-resolution proof of depth $\leq$ n exists. CoNLOGSPACE - hard to determine if an (all-positive, unit) proof of depth $\leq$ n exists.

d)	2LIT HORN, n unary	NLOGSPACE - complete to determine if an (all-positive, unit) proof of (depth, size) n exists.
e)	2LIT HORN, DET, n unary	DLOGSPACE - complete to determine if an (all-positive, unit) proof of (depth, size) n exists.
f)	arbitrary set of clauses, #P $\leq$ c, n unary	Regular set to determine if an (all-positive, all-negative, unit, hyper-resolution, arbitrary) proof of depth n exists.
g)	arbitrary set of clauses, #P $\leq$ c, n binary	Regular set to determine if an (all-positive, all-negative, unit, hyper-resolution,arbitrary) proof of depth n exists.

Comments: There is a polynomial time algorithm for the first part of a) even though the straightforward search method may take exponential time. See [4] for a related result. Also, the second part of a) is obtained by reduction from the hitting set problem [5]. Furthermore, results a) through e) can also be obtained for E = 2 and #P $\leq$ c and U = 0, and for E = 2 and #P $\leq$ c and U $\leq \log_2|W|$. This is done by encoding n predicate symbols as sequences of logn bits.

Note that the results in Part 5 give a precise meaning to the statements "unit resolution is easier than general resolution" and "all-positive resolution for Horn sets is easier than all-negative resolution for Horn sets".

6. Complexity of deciding the existence of a depth n resolution proof of NIL (the empty clause) from a set W of clauses, not necessarily in Schönfinkel-Bernays form. Let $|x|$ be the length of the input x consisting of W and n. The following results all still hold if the function symbols of W are restricted to a fixed finite set of at least two unary function symbols and at least one constant.

a) n binary — Complete for nondeterministic time $2^{2^{c|x|}}$. Still true if 3LIT HORN is required, and if we restrict the proofs to be all-positive, unit, or hyper-resolution proofs.

b) n unary — Complete for nondeterministic time $2^{c|x|}$. Still true if 3LIT HORN is required, et cetera as in part a).

c) 2LIT, n binary — Complete for nondeterministic time $2^{2^{c|x|}}$. Still true if HORN is required. Complete for NEXPTIME if we restrict the proofs to be all-positive or unit proofs, even if HORN is required.

d) 2LIT, n unary — Complete for NEXPTIME (i.e., nondeterministic time $2^{c|x|}$). Still true if HORN is required. NP-complete if we restrict the proofs to be all-positive or unit resolution proofs, even if HORN is required.

e) 3LIT HORN, DET, n binary — Complete for CoNEXPTIME to decide if a hyper-resolution proof of depth n exists. Hard for CoNEXPTIME to decide if an (all-positive, unit) proof of depth n exists.

f) 3LIT HORN, DET n unary — CoNP - complete to decide if a hyper-resolution proof of depth n exists. CoNP hard to decide if an (all-positive, unit) proof of depth n exists.

g) 2LIT HORN, DET n binary — Complete for deterministic time $2^{2^{c|x|}}$ to determine if an arbitrary resolution proof of depth n exists. Complete for DEXPTIME to determine if an (all-positive, unit) proof of depth n exists.

h)	2LIT HORN, DET n unary	Complete for DEXPTIME to determine if an arbitrary resolution proof of depth n exists. Complete for P (polynomial time) to determine if an (all-positive, unit) proof of depth n exists.

Comments: To obtain the upper bounds, we represent literals as directed acyclic graphs as in [10]. If L is such a literal, let $|L|$ denote the number of nodes in L. The number of edges in L will be proportional to $|L|$ if we have a fixed number of function symbols of fixed arities. In the worst case, the number of edges will still be $O(|L|^2)$. It may take $O(|L|^2 \log|L|)$ space to represent L in a Turing machine since each node may require $O(\log|L|)$ bits. We obtain the upper bounds by observing that if substitution σ is a most general unifier of literals A and B, then $|A\sigma| \leq |A| + |B|$ and $|B\sigma| \leq |A| + |B|$. Also, if C is a resolvent of C1 and C2 then $|C| \leq |C1| + |C2|$. The lower bounds are obtained by encoding Turing computations using two stacks. These results show that resolution theorem proving is inherently difficult, even when the depth is restricted. Also, these results suggest that resolution theorem provers should concentrate more on the "size" of a proof than on its depth, since proofs of small depth can have a large size. By size we mean in this context the sum of the sizes of the clauses in the proof. We know that finding proofs of small size is no worse than NP-complete.

Encoding the Successor Relation

Many of the completeness results for Schönfinkel-Bernays form are based on an economical encoding of the successor relation. We represent

integers in binary notation as sequences of zeroes and ones. Assume constants of zero and one and other constants are available. (We have restricted E to 2 so there are strictly speaking only two constants, but other constants can be encoded as sequences of these two.) Let $I(c_0, c_1, \ldots, c_n)$ be the integer $c_0 \cdot 2^n + c_1 \cdot 2^{n-1} + \ldots + c_{n-1} \cdot 2 + c_n$. We construct a formula A(P1, P2) in Schönfinkel-Bernays form which represents the following assertion:

$$\forall x_1 \forall x_2 \ldots \forall x_m [(0 \leq I(c_0, c_1, \ldots, c_n) < 2^{n+1}-1 \text{ and}$$
$$I(d_0, d_1, \ldots, d_n) = I(c_0, c_1, \ldots, c_n) + 1) \text{ implies}$$
$$P1(c_0, c_1, \ldots, c_n, x_1, x_2, \ldots, x_m) \equiv P2(d_0, d_1, \ldots, d_n, x_1, x_2, \ldots, x_m)].$$

Thus we can count up or down by one by going from P1 to P2 or from P2 to P1. Also, the formula A(P1, P2) consists of 2-literal Horn clauses and satisfies the restriction DET. The formula A(P1, P2) makes use of auxiliary predicate symbols Q1 and Q2, and is constructed as follows: Let c be a constant symbol other than 0 and 1. Let $\bar{x}$ be an abbreviation for $x_1, x_2, \ldots, x_m$, $\bar{y}$ be an abbreviation for $y_1, y_2, \ldots, y_n$, and $\bar{z}$ be an abbreviation for $z_1, z_2, \ldots, z_n$.

1.	$P1(\bar{y}, \bar{x}) \supset Q1(c, \bar{y}, \bar{x})$	Insert c
2.	$Q1(\bar{z}, 1, \bar{x}) \supset Q1(0, \bar{z}, \bar{x})$	Change rightmost ones to zeroes
3.	$Q1(\bar{z}, 0, \bar{x}) \supset Q2(1, \bar{z}, \bar{x})$	Change rightmost zero to one
4.	$Q2(\bar{z}, 1, \bar{x}) \supset Q2(1, \bar{z}, \bar{x})$	Cycle past remaining bits
5.	$Q2(\bar{z}, 0, \bar{x}) \supset Q2(0, \bar{z}, \bar{x})$	Cycle past remaining bits
6.	$Q2(\bar{z}, c, \bar{x}) \supset P2(\bar{z}, \bar{x})$	Delete c

Example: $P1(011\bar{x}) \supset Q1(c011\bar{x}) \supset Q1(0c01\bar{x}) \supset Q1(00c0\bar{x}) \supset Q2(100c\bar{x}) \supset P2(100\bar{x})$.

We represent W1 $\supset$ W2 by $\neg$W1 $\vee$ W2 and thereby obtain 2-literal Horn clauses. The above six clauses add one in going from P1 to P2. There are six similar clauses which subtract one in going from P2 to P1. The formula A(P1, P2) is the conjunction of these twelve clauses. Note that adding one always requires a constant number of steps, regardless of the number involved. This is useful for establishing bounds concerning sizes of resolution proofs.

Possible Extensions

These results can be extended in many ways. Some gaps in the above tables of results still exist. We might look at other classes of formulae, also. For example, we can consider the satisfiability problem for formulae in which the connectives are chosen from some set other than $\wedge$, $\vee$, $\neg$. This is related to recent work of Lewis [9]. We can consider other decidable subclasses of the first-order predicate calculus investigated by Lewis [7] with the HORN, 2LIT, or DET restrictions added. Also, we can consider decidable subclasses of the second order predicate calculus. We have shown an analogue of 3-satisfiability to be complete for nondeterministic exponential time. Are there analogues of other NP-complete problems (such as the clique problem) that are complete for nondeterministic exponential time? Many of our results are based on encodings of deterministic and nondeterministic Turing computations by predicate calculus formulae. Can parallel and alternating Turing computations also be modeled in this way? Lewis [7] has done some work along this line. Finally, we can look at inference rules other than resolution and determine the complexity of searching for proofs at restricted depths.

References

[1] Cook, S. A. The complexity of theorem proving procedures. Proc. 3rd Annual ACM Symp. on Theory of Computing (1971) 151-158.

[2] Hartmanis, J., Immerman, N. and Mahaney, S. One-way log tape reductions. Proc. 19th Annual Symp. on Found. of Computer Science (1978) 65-71.

[3] Jones, N. D., Lien, Y. E. and Laaser, W. T. New problems complete for nondeterministic log space. Math. Systems Theory 10 (1976) 1-18.

[4] Jones, N. D. and Laaser, W. T. Complete problems for deterministic polynomial time. Theoretical Computer Science 3 (1976) 105-117.

[5] Karp, R. M. Reducibility among combinatorial problems, in Miller, R. E. and Thatcher, J. W. (eds.), Complexity of Computer Computations (Plenum Press, New York, 1972) 85-103.

[6] Kozen, Dexter. Lower bounds for natural proof systems. Proc. 18th Annual Symp. on Found. of Computer Science (1977) 254-266.

[7] Lewis, Harry R. Complexity of solvable cases of the decision problem for the predicate calculus. Proc. 19th Annual Symp. on Found. of Computer Science (1978) 35-47.

[8] Lewis, Harry R. Personal communication.

[9] Lewis, H. R. Satisfiability problems for propositional calculi, unpublished manuscript (1978).

[10] Paterson, M. S. and Wegman, M. N. Linear unifications. Proc. 8th Annual ACM Symp. on Theory of Computing (1976) 181-186.

[11] Stockmeyer, L. J. and Meyer, A. R. Word problems requiring exponential time. Proc. 5th Annual ACM Symp. on Theory of Computing (1973) 1-9.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-79-978	2.	3. Recipient's Accession No.
4. Title and Subtitle Complete Problems in the First-Order Predicate Calculus			5. Report Date July 1979 6.
7. Author(s) David A. Plaisted			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Department of Computer Science University of Illinois Urbana, IL 61801			10. Project/Task/Work Unit No. 11. Contract/Grant No. NSF MCS 77-22830
12. Sponsoring Organization Name and Address National Science Foundation Washington, D.C.			13. Type of Report & Period Covered 14.

15. Supplementary Notes

16. Abstracts

We present a survey of recent results concerning the complexity of deciding if a first-order predicate calculus formula in Schönfinkel-Bernays form is satisfiable. We also present results concerning the complexity of deciding the existence of resolution proofs of restricted depths and sizes. In this way we obtain natural problems complete for the classes NP, PSPACE, deterministic and nondeterministic exponential time, and deterministic and nondeterministic double exponential time. The results concerning the existence of resolution proofs of restricted depths and sizes seem to have implications for the design and analysis of resolution theorem proving programs. This report mainly lists results without proof.

17. Key Words and Document Analysis. 17a. Descriptors

Satisfiability, predicate calculus, resolution theorem proving, complete problems, comptuational complexity.

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page) UNCLASSIFIED	22. Price